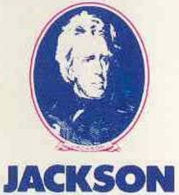


20 LECCIONES DE BASIC
PARA APRENDER CON EL SPECTRUM

INGELEK



JACKSON

- El casete
- Los soportes magnéticos
- Header: los datos en cinta
- Uso correcto del casete
- LOAD, SAVE, VERIFY MERGE, DIM
- Vectores y matrices
- Videoejercicios
- Videojuegos N.º 8



Spectrum

16K/48K/PLUS

VIDEO BASIC
Una publicación de
INGELEK JACKSON
Director editor por INGELEK:
Antonio M. Ferrer
Director editor por JACKSON HISPANIA:
Lorenzo Bertagnolio
Director de producción:
Vicente Robles
Autor: Softidea
Redacción software italiano:
Francesço Franceschini,
Stefano Cremonesi
Redacción software castellano:
Fernando López, Antonio Carvajal,
Alberto Caffarato, Pilar Manzanera
Diseño gráfico:
Studio Nuovaidea
Ilustraciones:
Cinzia Ferrari, Silvano Scolari,
Equipo Galata
Ediciones INGELEK, S. A.
Dirección, redacción y administración,
números atrasados y suscripciones:
Avda. Alfonso XIII, 141
28016 Madrid. Tel. 2505820
Fotocomposición: Espacio y Punto, S. A.
Imprime: Gráficas Reunidas, S. A.

ISBN del tomo 2: 84-85831-17-9
ISBN del fascículo: 84-85831-11-X
ISBN de la obra completa: 84-85831-10-1
Depósito Legal: M-15076-1985
Plan general de la obra:
20 fascículos y 20 casetes, de aparición quincenal,
coleccionables en 5 estuches.
Distribución en España:
COEDIS, S. A.
Valencia, 245. 08007 Barcelona.
INGELEK JACKSON garantiza la publicación de todos
los fascículos y casetes que componen esta obra y el
suministro de cualquier número atrasado o estuche
mientras dure la publicación y hasta un año después de
terminada.
El editor se reserva el derecho de modificar
el precio de venta del fascículo,
en el transcurso de la obra, si las circunstancias del
mercado así lo exigen.
Julio, 1985.
Impreso en España.

INGELEK



JACKSON

SUMARIO

Introducción

Una grabadora normal o casi normal es el periférico de memoria por excelencia. Sencilla de manejar y barata, la grabadora es capaz de escribir, leer y guardar de forma permanente en una cinta (un casete normal de audio) todas las informaciones, sean éstas programas o datos.
La térnica empleada es la misma que en grabaciones de audio normales pero con una única diferencia: las «notas» sólo son dos, 0 y 1.
Además existen instrucciones ligadas a la grabadora, que permiten el «diálogo» (enviar o recibir datos entre el casete y el ordenador), son: SAVE, VERIFY, LOAD y MERGE.

La grabadora

Un sistema para la elaboración de datos requiere, como condición indispensable para el tratamiento de la imformación, la presencia de dispositivos capaces de introducir datos en el sistema y de grabarlos a la salida de este. Como bien sabes, estas funciones las realizan las unidades llamadas de entrada/salida, que



están conectadas directamente al sistema y que son capaces de proporcionar o devolver, en su caso, lo que necesita la unidad central para ejecutar las diversas operaciones.
Entre todos los periféricos de entrada/salida, la grabadora es el dispositivo de empleo más común, tanto entre pequeños como entre grandes ordenadores.
Ya hemos dicho que la memoria central de lectura/escritura (es decir, RAM) posee en el momento actual un coste relativamente elevado (a pesar de los natables adelantos obtenidos durante los últimos tiempos en el campo de la tecnología electrónica), y este factor es el que normalmente obliga a los usuarios a emplear memorias RAM de capacidad limitada.
Además, ésta casi siempre suele ser volátil, es decir, que pierde irremediablemente su contenido cuando se apaga el ordenador. Por lo tanto, es necesario que los programas y los datos se puedan conservar en memorias no volátiles, llamadas también memorias de masa, precisamente porque son capaces de contener grandes cantidades de datos e informaciones.

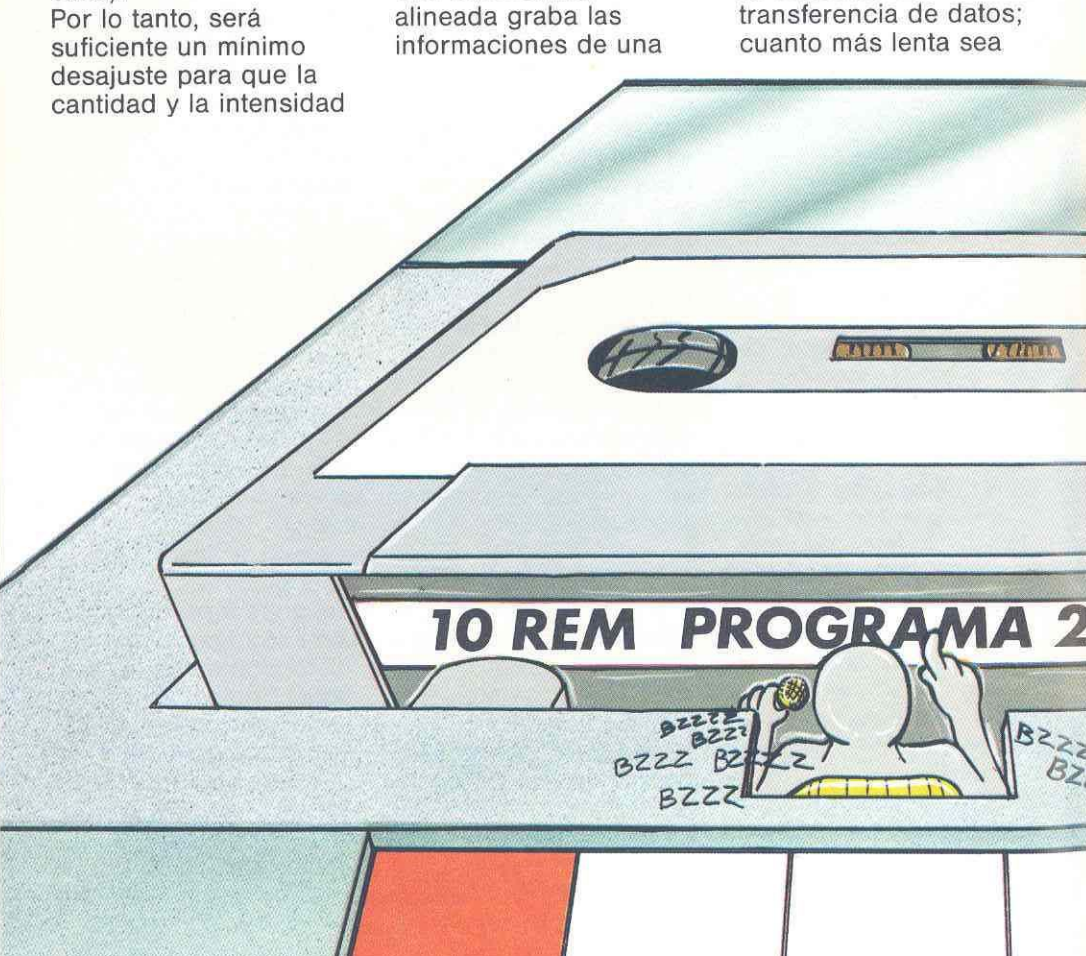
Los soportes magnéticos

El casete, o mejor dicho, la cinta magnética, es precisamente una memoria de masa; representa un soporte de grabación fundamental para los ordenadores, y se usa tanto como unidad de entrada como de salida, para memorizar definitivamente programas y resultados de cálculos, o, más sencillamente datos e informaciones.
Las unidades de cinta magnética ofrecen la posibilidad de introducir datos a la memoria central, con una velocidad suficientemente elevada, o la de grabarlos desde la salida en una forma eficaz, segura y, sobre todo, barata.
La tecnología y los principios de funcionamiento son idénticos para cualquier tipo de unidad de cinta, partiendo de los casetes hasta llegar a los superterminales de cinta de los principales centros de cálculo.
A la base de todo ello se encuentra la misma filosofía tecnológica,

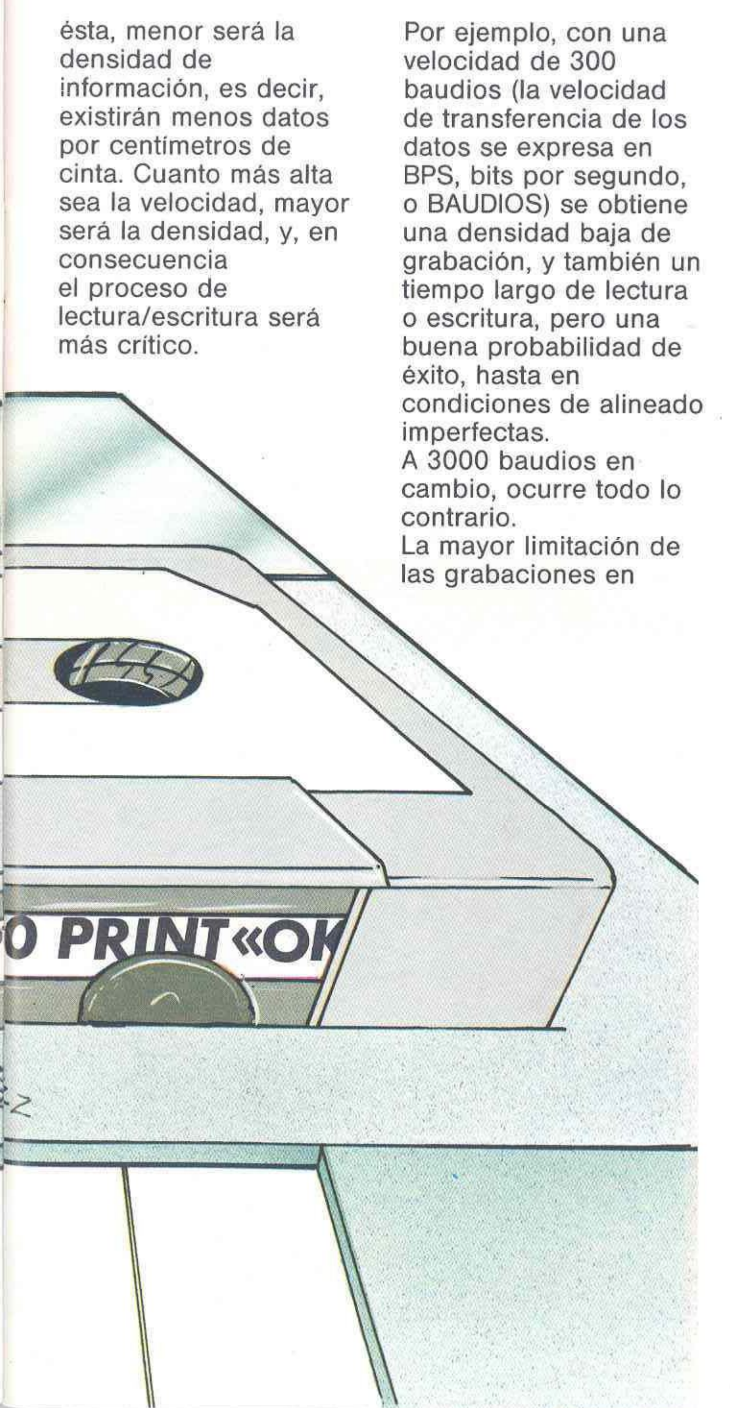
aunque aplicada, como es lógico, sobre escalas, proporciones y tecnologías bastante distintas.
En nuestro curso haremos referencia constante y explícita a las grabadoras normales de casete; aunque, a la vista de lo dicho, todo este discurso es aplicable y generalizable a cualquier otro tipo de dispositivo de cinta magnética.
En primer lugar, nos vamos a ocupar del principio sobre el que se basa la grabación magnética. Sin entrar en una docta disertación sobre magnetismo y electromagnetismo, digamos que algunos materiales, llamados ferromagnéticos, tienen la propiedad de imantarse en forma permanente cuando se les expone a un campo magnético intenso.
Este estado cesa (es modificado o anulado) en presencia de otro campo magnético suficientemente intenso.
Y llegamos a nuestro casete. La cinta magnética empleada está constituida por un soporte no magnético —una cinta flexible, fina y resistente de material plástico— sobre el que se ha depositado una capa extremadamente fina de sustancias magnéticas especiales (óxidos de hierro u otros), semejantes a numerosísimos imanes microscópicos, independientes los unos de los otros.
La cinta se hace pasar a velocidad constante por una «cabeza de grabación magnética», cuya función es la de transformar las señales eléctricas de entrada en sus correspondientes señales magnéticas.
Las variaciones del campo magnético producidas por el cabezal se traducen en invisibles movimientos de orientación de cada uno de los «imanes», que componen la capa sensible. Una vez que cesa el efecto del cabezal de grabación, estas partículas no logran volver a tomar su posición original, quedando así grabadas en la cinta de forma permanente las informaciones magnéticas transmitidas originariamente al casete a través de una señal eléctrica. Se ha efectuado así la grabación o escritura de la cinta.
La reproducción, o lectura, se obtiene en cambio haciendo pasar la cinta delante de una cabeza de lectura, de uso y funcionamiento muy similar básicamente al de la de grabación.
Al pasar delante de la cabeza, las partes magnéticas de la cinta provocan variaciones

del campo magnético iguales y contrarias a aquéllas que las habían generado, determinando, por lo tanto, sobre la cabeza que las percibe la aparición de una serie de señales eléctricas muy débiles.
Amplificándolas millares de veces se obtiene una señal muy similar a aquélla grabada.
Una importante particularidad de la grabación magnética es la posibilidad de borrar rápidamente la información almacenada, lo que permite reutilizar la cinta hasta centenares de veces, sin provocar deterioros apreciables en la calidad.
El método habitualmente empleado es el de hacer pasar la cinta magnética sobre una «cabeza de borrado», que produce un campo magnético muy intenso y «desordenar» todas las partículas de la capa sensible, borrando así todo lo que hubiera sido memorizado anteriormente.
El funcionamiento de este cabezal es aproximadamente igual al de la cabeza grabadora, con la única diferencia de que se le aplica una señal eléctrica constante y de una frecuencia elevadísima, generada por un circuito de la grabadora.
En las grabadoras normales la cabeza de borrado está situada al lado de la cabeza de grabación, de tal manera que en el momento de grabar algo, la cinta sea obligatoriamente borrada, quedando así en las mejores condiciones antes de iniciarse el proceso de grabación.
Naturalmente, son de fundamental importancia en todo el proceso tanto la velocidad de arrastre de la cinta (que ha de mantenerse lo más constante, exacta y precisa posible), como la zona de contacto entre el cabezal y la cinta magnética (que siempre debe conservar una determinada alineación).
Una grabadora que no respetara alguna de estas normas, nunca podría desempeñar correctamente su trabajo, es decir, memorizar las informaciones y después reproducirlas sin introducir ninguna distorsión sobre la señal original y, por tanto, sin errores.
La naturaleza de los posibles errores puede ser muy variada: desde un error general de carga hasta la interrupción a mitad del programa, superar la capacidad de memoria disponible, etc. El

problema, como ya se ha señalado, se deberá en la mayoría de los casos a un mal arrastre de la cinta (velocidad no constante), y sobre todo al mal alineamiento del cabezal (no perpendicular a la cinta).
Por lo tanto, será suficiente un mínimo desajuste para que la cantidad y la intensidad de la señal de lectura disminuya drásticamente.
Se trata de un efecto bastante traicionero, puesto que no es comprobable mediante datos guardados con el mismo casete.
Una cabeza mal alineada graba las informaciones de una forma francamente «personal»; pero, al igual que las graba es capaz de releerlas, dando la sensación de un funcionamiento correcto.
Además, los problemas de grabación aumentan la velocidad de transferencia de datos; cuanto más lenta sea



ésta, menor será la densidad de información, es decir, existirán menos datos por centímetros de cinta. Cuanto más alta sea la velocidad, mayor será la densidad, y, en consecuencia el proceso de lectura/escritura será más crítico.



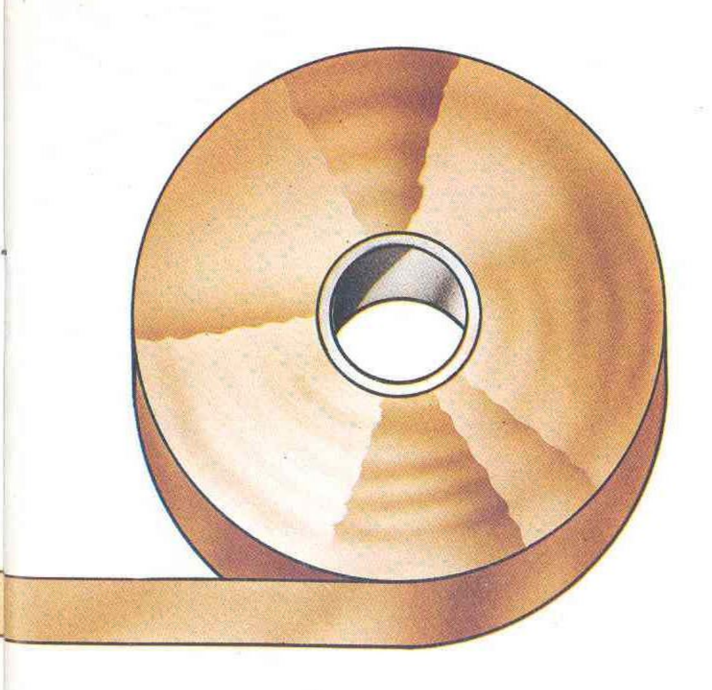
Por ejemplo, con una velocidad de 300 baudios (la velocidad de transferencia de los datos se expresa en BPS, bits por segundo, o BAUDIOS) se obtiene una densidad baja de grabación, y también un tiempo largo de lectura o escritura, pero una buena probabilidad de éxito, hasta en condiciones de alineado imperfectas.
A 3000 baudios en cambio, ocurre todo lo contrario.
La mayor limitación de las grabaciones en cinta magnética se debe precisamente al soporte empleado, puesto que una cinta para ser grabada o leída debe correr desde el principio hasta el final. Por lo tanto, su uso únicamente puede ser secuencial, es decir, escribiendo datos e informaciones uno tras otro.
En la base de lectura estos datos se tomarán siguiento la misma secuencia; será posible buscar datos que interesen durante el paso de la cinta, pero no será posible tomar informaciones que hayan pasado ya. Para recuperar un dato anterior será necesario rebobinar la cinta e iniciar su búsqueda desde el principio.
A pesar de todo esto, el casete es de uso extremadamente común: su comodidad y la sencillez de su funcionamiento, unidas a su notable fiabilidad y economía, le sitúan entre las unidades de memorización más populares.
De cualquier forma en las próximas lecciones veremos como existen otros dispositivos que eluden las limitaciones indicadas.

Header: los datos en cinta

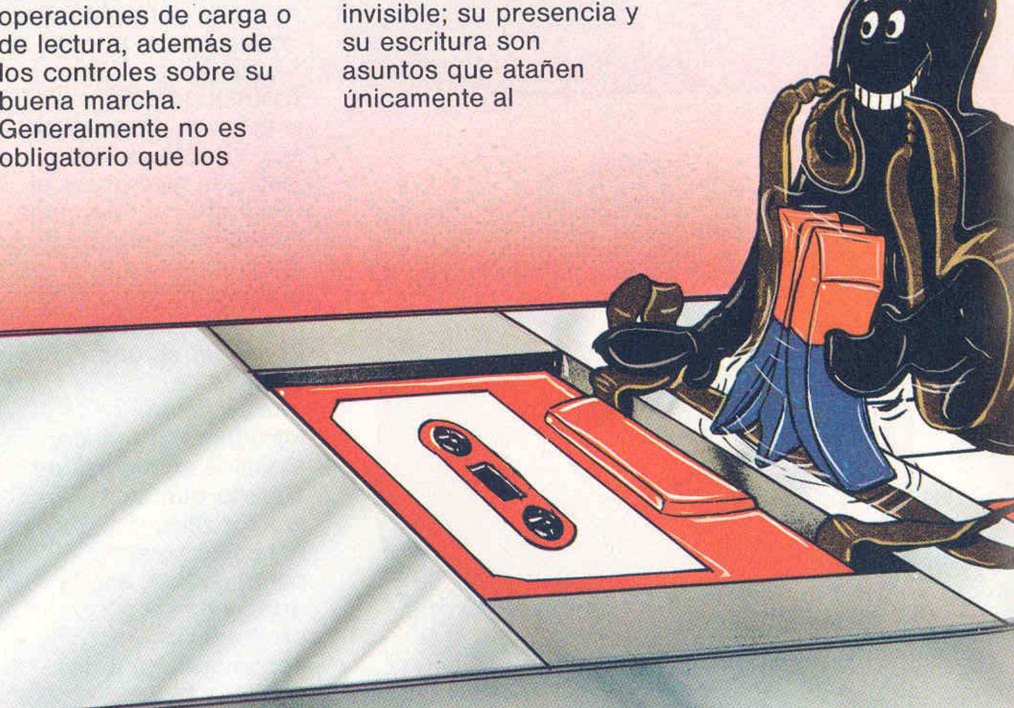
Ahora que has aprendido lo principal sobre el funcionamiento de la grabadora, podemos pasar a la segunda parte de nuestra lección, que hace referencia a su conexión y «diálogo» con tu ordenador. Expliquemos en primer lugar las formas en que los ordenadores memorizan y leen los datos, las informaciones y los programas a través de las cintas magnéticas.
Como es costumbre, cuando se realiza una conexión entre la unidad central y un periférico, se necesitan dos cosas indispensables: un interface (que permita a la CPU comunicarse con el exterior) y una conexión (que una físicamente el ordenador al periférico). El ordenador no escapa a este regla.
De cualquier forma, además de la conexión física, es necesario que entre las dos unidades exista también un perfecto «entendimiento» acerca de la comunicación, es decir, una regla para el diálogo y la recepción de las informaciones, tanto en entrada como en salida. El asunto tiene apariencia de ser complicado, pero intentaremos aclararlo con un ejemplo que nos

PUNTOS MAGNETIZADOS

enseñe qué es lo que ocurre cuando se le imparte a la unidad central la orden de memorizar un programa en cinta.
Lo primero que comprueba la CPU es que el depósito de recepción, es decir, el casete, esté preparado para la llegada de las informaciones. Si todo es correcto, puede empezar la transferencia del programa a grabar. Esta operación se ejecuta mediante el interface adecuado, que transmite los datos a través de una conexión en serie, enviando al casete un bit detrás del otro.
¿Por qué en serie? La respuesta es bien sencilla. El casete memoriza las informaciones que llegan en un orden secuencial, es decir, la una a continuación de la otra.
Por lo tanto, es natural que se elija como estándar de conexión y comunicación aquél que más se acerca a este modo de operar, es decir, aquél que posee la misma característica de secuencialidad.
Terminada la transmisión se encuentra sobre la cinta la versión «magnética» del programa –cada bit está representado por la presencia o ausencia de información magnética– que está ya lista para ser volcada nuevamente, cuando sea necesario, a la memoria del ordenador.
En todo este proceso se ha omitido voluntariamente una acción muy importante que ejecuta la CPU antes de memorizar el programa: la operación previa de escribir sobre la cinta, al comenzar la grabación, el llamado *header* (cabecera).
El *header* es una especie de presentación introductoria *(header* significa cabecera) de aquello que será memorizado inmediatamente a continuación; así, la grabación de un programa es encomendada a dos distintos bloques separados entre ellos



por un breve espacio. El primero es un bloque preliminar, que contiene el nombre y alguna otra información sobre el programa; el segundo comprende en cambio el programa en sí.
La presencia y función de la cabecera, que normalmente empieza con una nota continua, son fundamentales: le comunica a la unidad central el nombre del programa, su longitud y otras informaciones que permiten la sincronización de las operaciones de carga o de lectura, además de los controles sobre su buena marcha.
Generalmente no es obligatorio que los programas tengan un nombre, pero es aconsejable ponérselo para evitar confundir distintas informaciones.
La grabación de la cabecera la realiza automáticamente el ordenador (se encarga de ello el sistema operativo) durante la carga: por lo tanto, no se requiere ningún trabajo extra por parte del programador.
Más aún, respecto al usuario se puede decir que la cabecera es «transparente», es decir, invisible; su presencia y su escritura son asuntos que atañen únicamente al ordenador, de quien dependen.
De lo que hay que asegurarse es de que las lengüetas de protección contra escritura situadas en el borde de la cinta casete estén sin quitar o bloqueadas (permitiendo así la memorización) y de que la cinta no esté completamente rebobinada.
Normalmente, la cinta



va precedida por un breve tramo no magnetizado (fácilmente reconocible puesto que es transparente o rojo); si el encabezamiento fuera enviado a esta zona no grabable, la consecuencia sería que el programa que le sigue ya no podría ser recuperado, puesto que le faltaría a la CPU la parte de sincronización y reconocimiento.
Por lo tanto, ten cuidado.
De cualquier forma, para aquéllos que no quieran correr ningún riesgo, existen en el mercado casetes especiales, diseñados para el uso con ordenadores, a los que les ha sido eliminada esta parte «en blanco».
Por lo que respecta a la calidad de grabación hay que decir que ésta depende —aparte nuturalmente de la grabadora— del «grano» de la capa magnética de la cinta, es decir, del tamaño medio de cada una de las partículas magnetizables que constituyen el soporte magnético. Cuanto más fino sea el grano, tanto más capaz será la cinta de percibir las variaciones más rápidas de campo magnético.
Pero a diferencia de las grabaciones de audio (el oído humano percibe frecuencias entre 0 y 14.000 hertzios), el ordenador es un poco «sordo»: su máximo son 3.000 hertzios.
Por lo tanto, no es necesario gastarse una fortuna en equipos y materiales para obtener buenas grabaciones; las únicas normas a tener en cuenta son las de guardar los casetes en un lugar fresco y seco (y fundamentalmente, alejado de fuentes de campos magnéticos fuertes, como motores eléctricos, teléfonos o imanes), la de limpiar periódicamente, y siempre con productos adecuados, los cabezales de lectura/grabación, y la de hacer revisar de vez en cuando la grabadora en un taller especializado.
Un último consejo. No guardes ni cintas ni grabadora cerca del televisor: es una fuente de campos magnéticos intensos, que pueden modificar las grabaciones contenidas en las cintas.

Vectores y matrices

Hasta ahora hemos hablado únicamente de variables simples, es decir de variables que en un momento determinado de la ejecución contienen un único valor. Pero muchas veces resulta cómodo, o hasta necesario, poder tratar conjuntos de datos y variables ligados entre ellos por algún factor común. Por ejemplo: los días de la semana, los meses del año, o los apellidos de los participantes en una carrera.

Nuestra mente es capaz de tratar al mismo tiempo grandes cantidades de información, debido a que organiza, en estructuras más amplias, todos los términos asociados entre ellos. Por ejemplo, nadie se acuerda de las letras del alfabeto como de una serie de símbolos sin ninguna relación; más bien, hemos memorizado todas las letras en una única lista que empieza con la A. Si lo pones en duda, intenta enumerar las letras del alfabeto a partir de la Z, y con la misma rapidez que consigues pronunciándolas en el orden inicial: ¿sigues teniendo dudas?

También los ordenadores han sido proyectados y construidos para manipular grandes cantidades de variables sencillas, reagrupándolas por su similitud en estructuras de datos más amplias. Estas estructuras toman el nombre de vectores.

Un vector es una colección unidimensional ordenada de variables —a las que se puede hacer referencia empleando un mismo nombre —cada una de las cuales representa un dato o un valor relacionado con los demás a través de una ligazón lógica o relacional.

A un conjunto de variables de dos dimensiones se le denomina tabla y en general (para cualquier número de dimensiones) matriz.

Las variables del vector son llamadas elementos del vector y se distinguen las unas de las otras mediante su número de posición en el interior del vector mismo; por esta razón, este número es llamado índice o subíndice.

Un vector (o una tabla, o una matriz) puede ser numérica o alfanumérica, pero todos los elementos pertenecientes a un determinado vector deben forzosamente ser del mismo tipo que el del vector. No se permite que se mezclen datos de distinto tipo (y cualquier intento de hacerlo llevará inevitablemente a la obtención de un mensaje de error por parte del intérprete BASIC).

El nombre de un vector o de una matriz puede estar formado por una sola letra. Esto es válido tanto para los numéricos como para los alfanuméricos. Como de costumbre este último estará identificado por el símbolo $.
Los índices de los elementos deberán situarse entre paréntesis. Así, una expresión del tipo

```
PRINT A(17)
```

significa: imprime el elemento decimoséptimo perteneciente al vector A.

DIM

Antes de que podamos emplear una matriz, esta debe ser «declarada», es decir, hay que avisar al ordenador para que reserve un cierto número de localizaciones consecutivas en la memoria para guardar allí todos los elementos. La unidad central no puede saber previamente cuánto espacio de memoria le vas a requerir. Por lo tanto, para crear una tabla es necesario indicar con anterioridad el número de elementos que formarán parte de ella. Esto se consigue con la instrucción DIM (abreviatura de DIMensional); mediante esta instrucción se define el tamaño de la matriz y se reserva en memoria el espacio suficiente.

Ejemplos

```
DIM T(15)
```

Crea una matriz de 15 variables reales llamadas T.

```
T(1), T(2), T(3)......, T(12), T(13), T(14), T(15)
```

En el momento de indicar DIM se le atribuirá a cada elemento el valor nulo, es decir, 0 en el caso de matrices numéricas y la cadena nula para las matrices alfanuméricas.
Una vez que el vector haya sido definido, podrás utilizar sus elementos como si fueran cualquier otra variable.

```
A(13) = 5.3
```

Asigna al decimotercer elemento de la tabla el valor 5.3.

```
F(19) = F(19) - 3.17
```

Resta 3.17 al valor contenido en la decimonovena variable del vector F.

```
C$ (3) = "marzo"
```

Le asigna al tercer elemento de la matriz C$ (de tipo cadena) el valor "marzo".

Además, en lugar de emplear un número, es posible especificar el índice mediante una variable o expresión; en este caso se deberá poner un cuidado especial para que en ninguna circunstancia esta variable o expresión asuma valores ilícitos (mayores que la dimensión máxima o menores que 1).

```
LET K = 7
LET V (K) = 73
```

Asigna al séptimo elemento del vector V el valor 73.

```
LET V (K - 13) = 9
```

¡Error! Está fuera de los límites.
La posibilidad de representar al subíndice por medio de variables permite listar la totalidad de la matriz empleando un bucle:

```
 10 DIM V$ (7)
 20 LET V$(1 TO 7) =
      "VACACIONES"
 90 ...
110 FOR I = 1 TO 7
120 PRINT I V$ (I)
130 NEXT I
140 ...
```

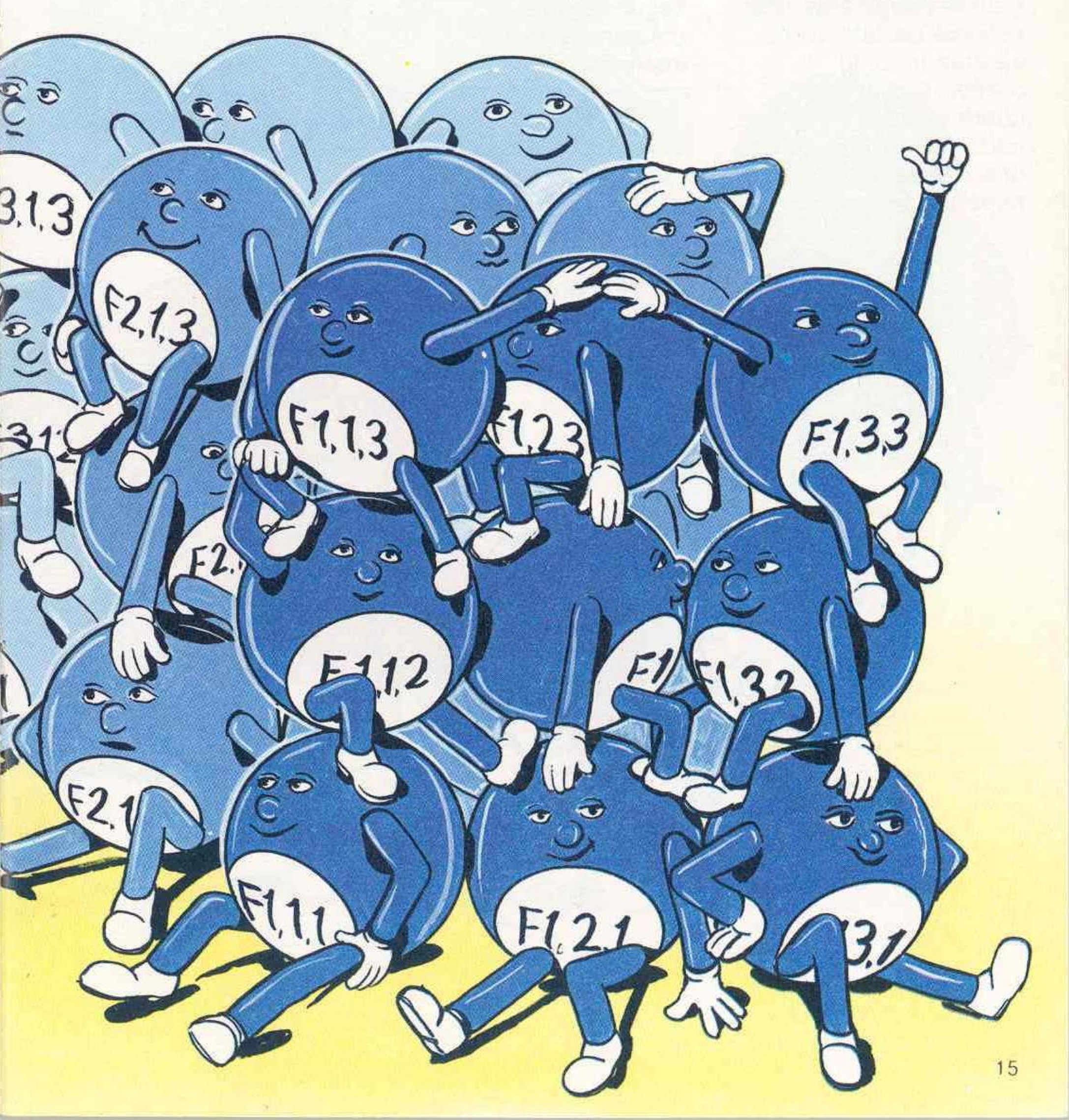


En el caso de que el índice contuviera una parte decimal se le redondea al valor entero más próximo.

R (2.8) equivale a R(3)

Por otra parte, las tablas no están limitadas a poseer un solo índice; se pueden tener matrices con dos, tres o más índices.

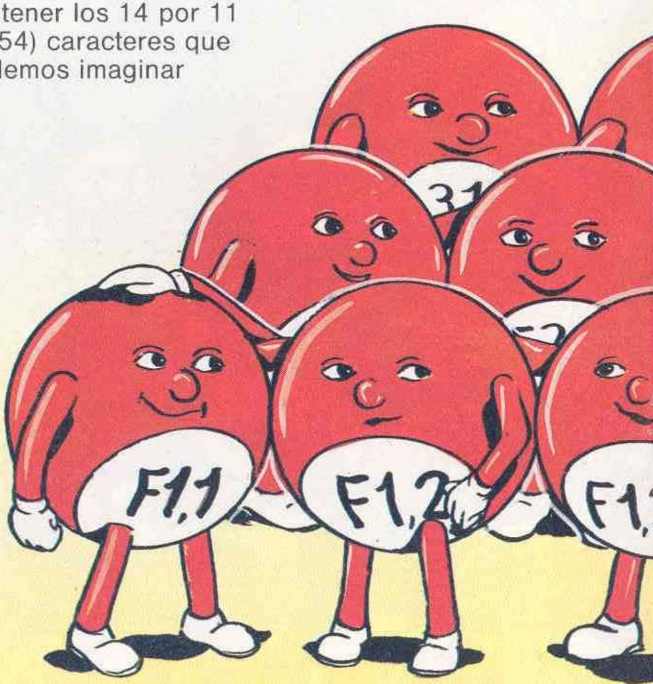
Esto no cambia nada en el interior de la memoria; se trata únicamente (como ya hemos comentado anteriormente) de una representación más cercana al modo de razonar de las



personas.
Por ejemplo, si piensas en la tabla de multiplicar, seguramente no tienes en la cabeza la representación de los números según una estructura unidimensional; más bien te imaginarás una especie de tablero de ajedrez (o, si lo prefieres, una hoja cuadriculada) en la que cada número ocupa una casilla: se trata pues de una representación bidimensional.
El BASIC te permite crear matrices dimensionadas con dos índices (separados por una coma): el primero hace referencia a las líneas de la matriz y el segundo a las columnas. Una instrucción como:

```
DIM Y$ (14,11)
```

reserva en la memoria de tu ordenador un espacio suficiente para contener los 14 por 11 (=154) caracteres que podemos imaginar dispuestos en 14 líneas y 11 columnas.
Hay que poner especial atención en la distinción entre líneas y columnas: el elemento Y$(4,3) ocupa dentro de la matriz una posición bien distinta de la del elemento Y$(3,4), aunque sea posible que en un momento determinado estos elementos contengan el mismo valor.

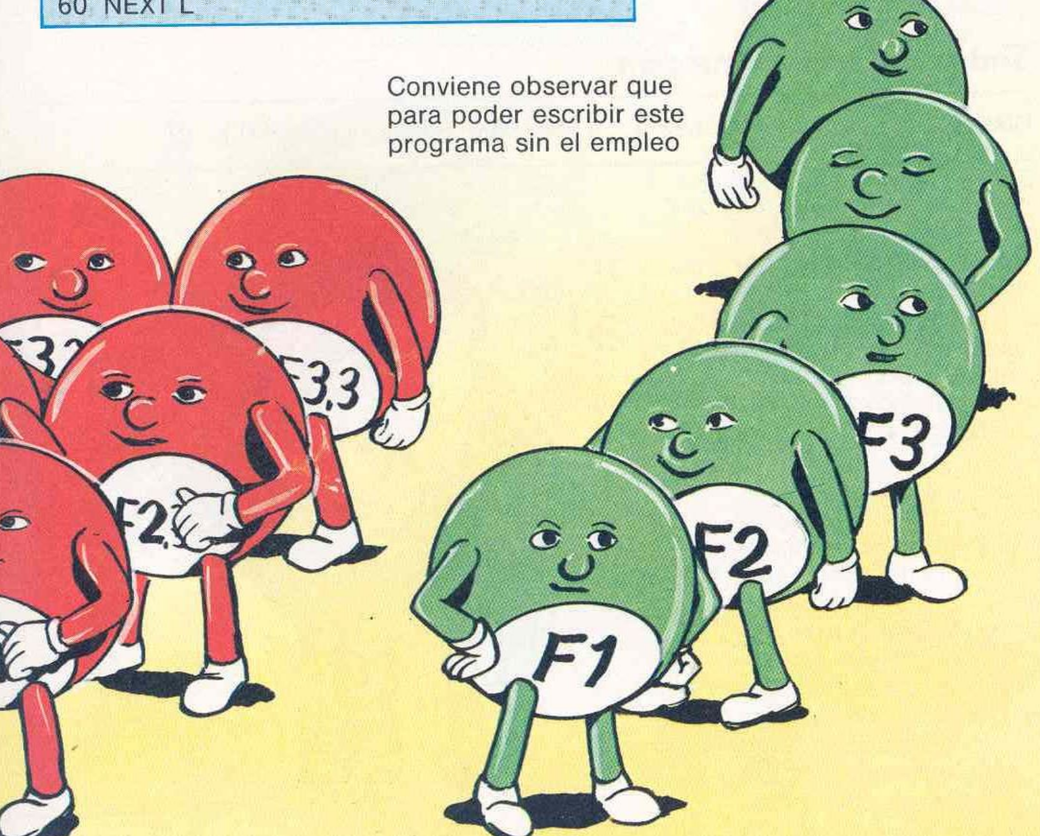


El breve programa siguiente carga en la memoria la tabla de multiplicar hasta 10.

```
10 DIM P (10,10):REM MATRIZ TABLA DE
   MULTIPLICAR
20 FOR L = 1 TO 10: REM L = LINEAS
30 FOR C = 1 TO 10: REM C = COLUMNAS
40 LET P (L, C) = L * C: REM VALOR DE LA
   LINEA L COLUMNA C
50 NEXT C
60 NEXT L
```

Conviene observar que para poder escribir este programa sin el empleo de matrices habría sido necesario emplear 100 variables simples, con todos los problemas y desventajas que esto conllevaría.

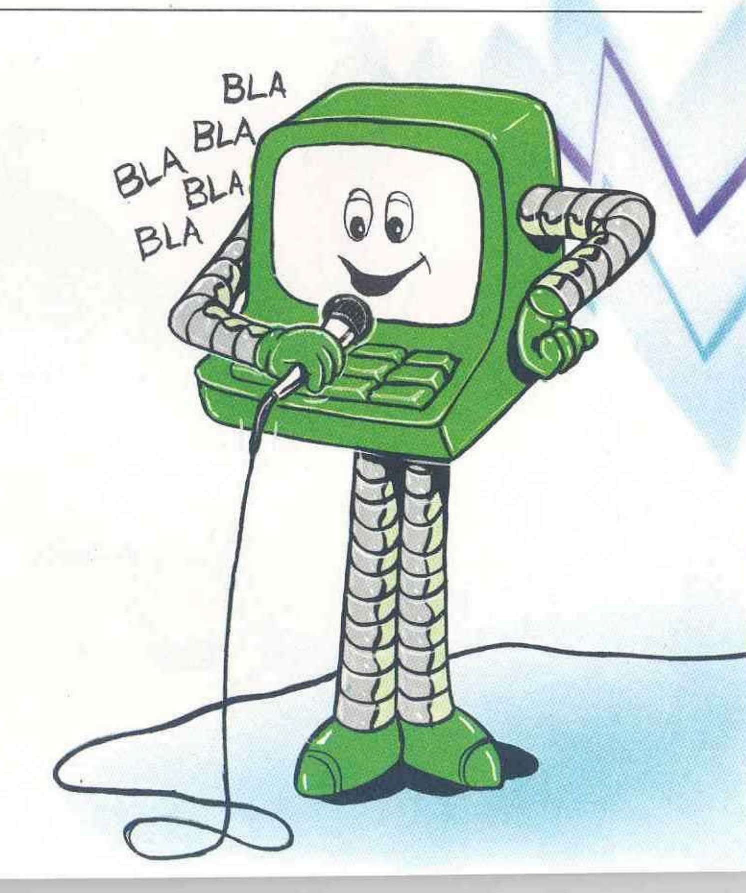
También se pueden dimensionar matrices con tres índices: sus elementos formarían



entonces una estructura tridimensional, que podríamos representar como un paralelepípedo en el espacio.
Las dimensiones más allá de la tercera escapan a nuestra capacidad de representación visual, y por esta razón su uso (por lo demás, totalmente permitido en BASIC) es raro y desaconsejable; resulta difícil mantener en la cabeza estos tipos de modelos.
En la lección de programación veremos prácticamente y con más profundidad todo lo que hemos comentado hasta ahora.

Sintaxis de la instrucción

DIM variable (índice1 [,índice2] [,...]) [,variable (índice1 [,índice2] [,...])]



SAVE

Tu Spectrum reconoce algunas instrucciones sencillas para manejar programas hacia y desde las memorias de masa: gracias a estas instrucciones es posible establecer un coloquio entre el ordenador y los periféricos de memoria (limitaremos de momento nuestra atención a un casete «normal», siendo nuestra intención volver sobre estas instrucciones en otro momento posterior, cuando analicemos otros periféricos de memoria).
Lo único necesario es pulsar algunas teclas del casete siguiendo las instrucciones que vayan apareciendo en pantalla.
Veamos, por lo tanto, cuáles son estas instrucciones, empezando por la fundamental (para los programadores, no para los «carga-dependientes»).
Como ya sabes, la memoria central de tu Spectrum pierde todo su contenido al ser desconectado de la alimentación. Por lo tanto, es necesario guardar un programa en una memoria exterior no volátil, como es la cinta, y esto para no tener que teclear cada vez el mismo programa.
Como siempre, es el BASIC quien se ocupa de esta importantísima función gracias a la instrucción adecuada: SAVE.
SAVE sirve para transferir a la cinta el programa contenido en la memoria de tu ordenador.
Al guardar, grabar, o «salvar» (del inglés, to save) un programa, se le da un nombre, de 10 caracteres como máximo, para después poder reconocerlo y realizar con él cualquier operación.
El envío de los datos



desde el ordenador al casete y viceversa se realiza en serie, es decir, 1 bit a la vez, limitándose sencillamente a copiar los datos de la memoria y enviándolos a la cinta a una velocidad de 1500 baudios (bit/seg).
Según el tiempo de grabación o carga, como después veremos, puedes averiguar la longitud del programa.
Sin que tú tengas que hacer nada, tu ordenador, o, mejor dicho, su sistema operativo, se ocupa de colocar al principio de la grabación el encabezamiento, que contiene los datos necesarios para el reconocimiento del programa por parte de la CPU, el nombre del programa, su longitud en bytes, y la primera localización que ocupa en la memoria.
Veamos ahora qué operaciones tienes que realizar para guardar un programa.
Una vez tecleado SAVE, pon en marcha el casete pulsando simultáneamente las teclas PLAY y REC; a continuación, pulsa una tecla cualquiera.
En pantalla aparecerá el mensaje.

```
START TAPE, THEN PRESS ANY KEY
```

(Si algo no funcionara, o, simplemente, desearas interrumpir la grabación, pulsa la tecla SPACE).
Apenas haya terminado, la grabación, visualizará el mensaje:

```
0 OK
```

He aquí ahora unos consejos dictados por la experiencia:

- nunca pienses que abusas de SAVE.
Guarda tus programas a medida que los vayas confeccionando.
Un contacto eléctrico defectuoso, un amigo inoportuno..., o un apagón, pueden hacerte perder un montón de horas de trabajo;
- Antes de grabar, asegúrate de que el sector de cinta que vas a utilizar no contenga otro programa, o parte de él, que pudiera serte útil; lo perderías para siempre.
- Usa varios casetes y preferiblemente cortos: algunos para el trabajo diario y otros para guardar aquéllos programas que te sean indispensables.
Si no deseas correr el riesgo de estropear una grabación, quita las lengüetas situadas en la parte trasera del casete.
Siguiendo todos estos consejos evitarás todos los inconvenientes con los que los programadores principiantes (incluyéndonos nosotros) hemos tropezado.
Un último comentario: también puedes guardar un programa escribiendo

```
SAVE "NOMBRE DEL PROGRAMA" LINE NUMERO
```

y ¿esto para qué sirve? Algo de suspense no hace daño: dentro de poco te desvelaré el arcano.

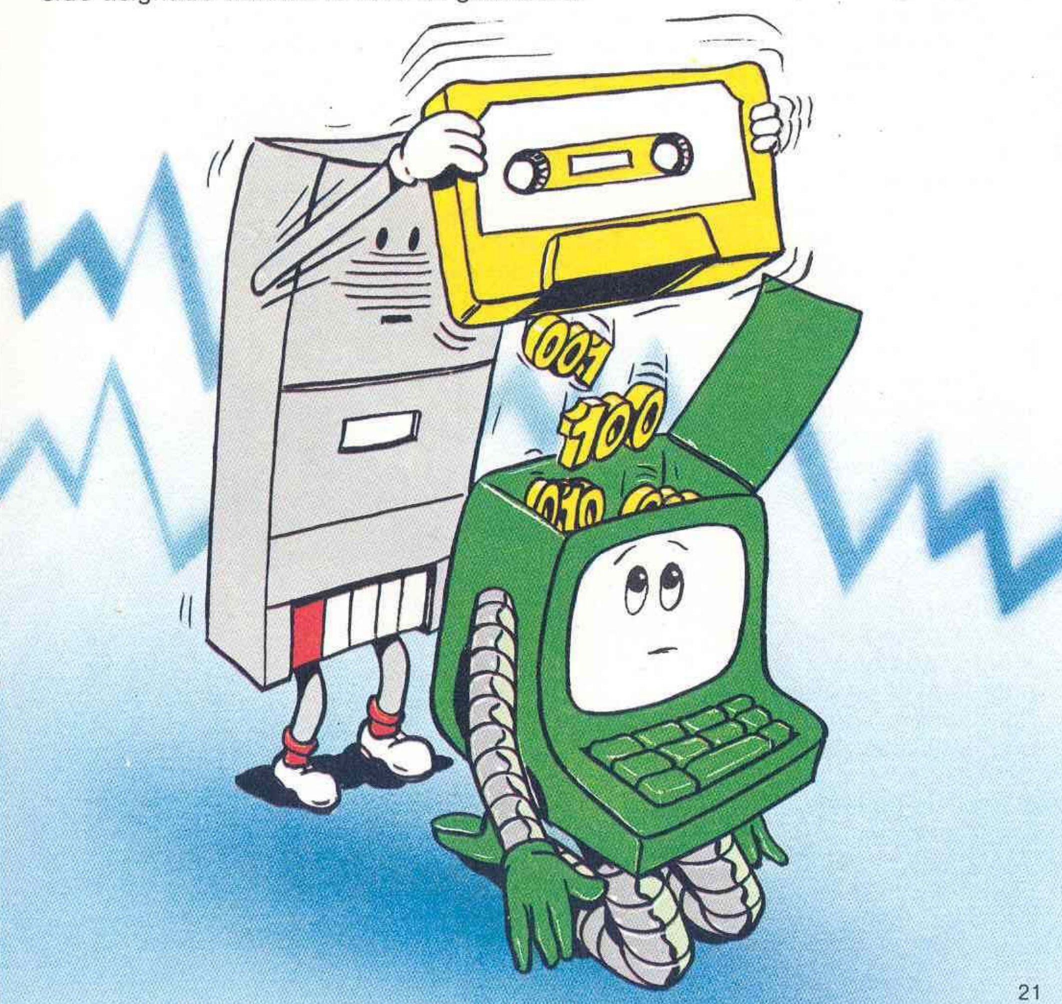
Sintaxis del comando/instrucción

SAVE "nombre del programa"

El nombre del programa es el nombre que le haya sido asignado durante la fase de grabación.

LOAD

Una vez que hayas guardado un programa, o que hayas adquirido un casete comercial, te encuentras con el problema (es un decir, puesto que sigues este curso) de cargarlo, es



decir, de efectuar la operación inversa a SAVE: transferir el programa desde la memoria de masa a la memoria central.
La instrucción, o comando, empleada para esta transferencia es LOAD; la operación recíproca de SAVE.
Como es natural, sirven para LOAD los mismos comentarios hechos para SAVE (velocidad de transferencia, barras horizontales,...) con una pequeña variación.
LOAD puede ser seguido por el nombre o por " " (dos comillas).
En el segundo caso, tu Spectrum cargará el primer programa que encuentre en la cinta.
En el primer caso efectúa la búsqueda del programa especificado por su nombre, ignorando los demás que encuentre por el camino.
Un aspecto básico: el nombre ha de ser absolutamente idéntico a aquél con el que fue guardado el programa.
Por ejemplo, si guardas un programa con: SAVE "MAMBRU 1" (observa el espacio en blanco) y luego indicas LOAD "MAMBRU 1", tu Spectrum buscará sin encontrar nada.

Veamos ahora qué es exactamente lo que ocurre.
Después de haber indicado, por ejemplo, LOAD "NUBES" tu Spectrum comenzará la búsqueda del programa "NUBES".
A medida que vaya encontrando programas visualizará el nombre con el que hayan sido guardados, continuando la búsqueda en el caso de que no sea el indicado.
Tras haber encontrado el programa es suficiente con teclear RUN para hacerlo correr (hablando en términos técnicos).

Nos queda una cuestión en suspenso desde que hablamos de SAVE, así que... desvelamos el arcano. Si el programa hubiera sido guardado con

```
SAVE "NUBES" LINE 3
```

una vez que el Spectrum lo encuentre, este programa arrancará automáticamente desde la línea indicada después de LINE.
Siguiendo el consejo de guardar con toda



tranquilidad que te hemos proporcionado al hablar de SAVE, siempre estarás seguro de encontrar inmediatamente el programa deseado. Una última advertencia: también aquí, al igual que al grabar un programa, es necesario poner atención puesto que LOAD borra de la memoria central el programa anterior y sus variables. Como siempre, no es mala una reflexión rápida sobre aquello que uno se dispone a hacer.

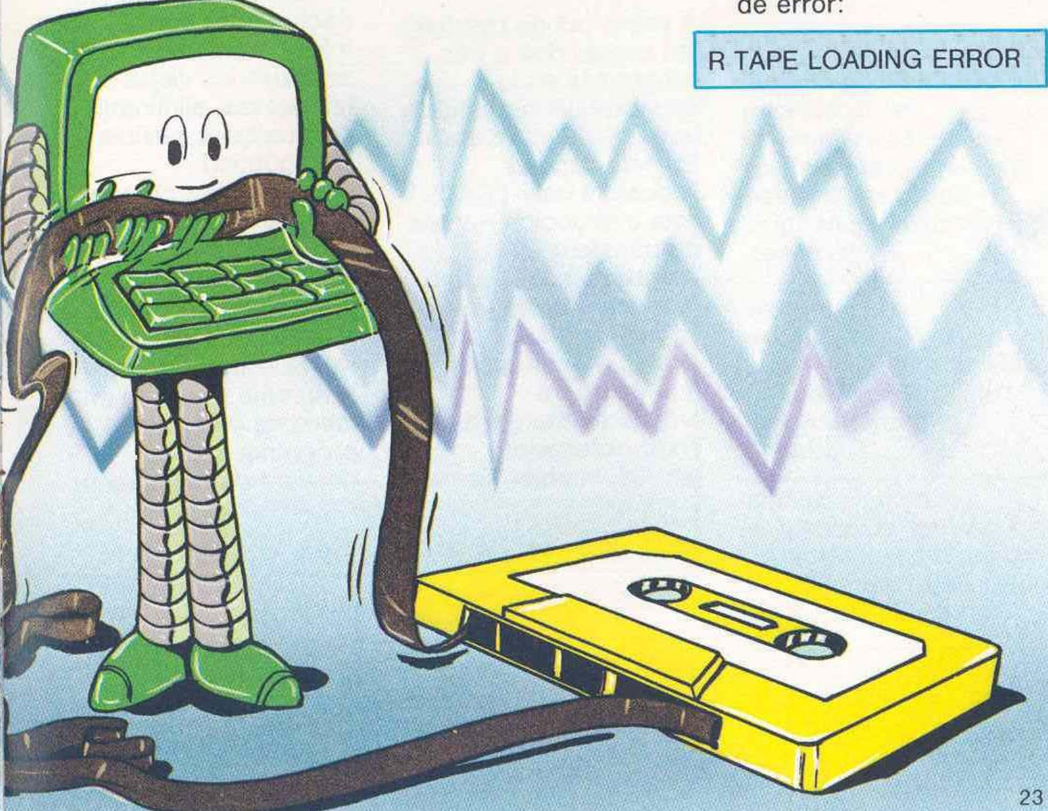
Sintaxis del comando/instrucción

LOAD "nombre del programa"

VERIFY

Para comprobar que después de una orden SAVE las cosas hayan funcionado correctamente (es decir, que no haya habido anomalías de grabación) es posible cotejar el programa en memoria con su copia correspondiente recién memorizada en la cinta. Si en esta comparación se detectan diferencias, obtenemos el mensaje de error:

```
R TAPE LOADING ERROR
```



Pero para poder hacer esta comprobación es necesario indicarle a tu Spectrum con qué debe de cotejar el programa que tiene en la memoria central.
Por esta razón VERIFY debe ser seguido por el nombre del programa del que haya que comprobar su correcta grabación.
Si no se indica un nombre, se comparará el programa en memoria con el primero que llegue desde la cinta.

Sintaxis de la instrucción

VERIFY "nombre del programa"

MERGE

A veces puede resultar útil cargar dos o tres programas en la memoria del ordenador, uniéndolos entre ellos.
En este caso es imposible usar LOAD: esta instrucción –antes de transferir el programa a la memoria– borra, como ya hemos dicho, cualquier instrucción que estuviera anteriormente presente en tu ordenador.
MERGE también carga un programa grabado en casete en la memoria del ordenador, pero a diferencia de LOAD, aún manteniendo su misma sintaxis, no borra el antiguo programa antes de iniciar la carga. Así pues, funde el programa especificado en la orden con las instrucciones que ya estuvieran presentes en la memoria.
En el caso de que existieran líneas en común entre el viejo y el nuevo programa (es decir, líneas con el mismo número), MERGE sustituye las viejas con las nuevas, eliminando así cualquier posible ambigüedad.
Con el mismo criterio que para SAVE y LOAD, si a MERGE no se le hace seguir ningún nombre de programa, la operación de fusión se realizará entre el programa presente en memoria y el primer programa que se encuentre en el casete.

Sintaxis de la instrucción

MERGE ("nombre del programa")

donde "nombre del programa" es, como de costumbre, el nombre del programa a tomar de la grabadora.

DATA, CODE

Las operaciones de transferencia de datos desde un ordenador a la cinta, o viceversa, pueden referirse además de a programas, también a otros tipos de datos (vectores, matrices, bytes). Sin embargo, para comunicarle al ordenador que se trata de informaciones distintas a un programa, es necesario añadir después de una instrucción LOAD, SAVE o VERIFY una de las siguientes palabras clave:
DATA si se trata de una matriz o de un vector;
CODE si se trata de informaciones en código máquina.
En este último caso, en el momento de guardar el programa desde el ordenador a la cinta, es indispensable especificar desde qué localización de memoria se transfieren los códigos, y cuantos son estos.

Ejemplos

```
SAVE "vocablos" DATA V$ ()
```

Guarda una tabla de informaciones alfanuméricas reconocida mediante el nombre «vocablos» y asignada a la variable de matriz V$

```
LOAD "números" DATA B()
```

Busca en la cinta la matriz «números», y una vez que la ha encontrado, si existe suficiente memoria libre, la transfiere al ordenador con el nombre B.

```
SAVE "música" CODE 29000, 1000
```

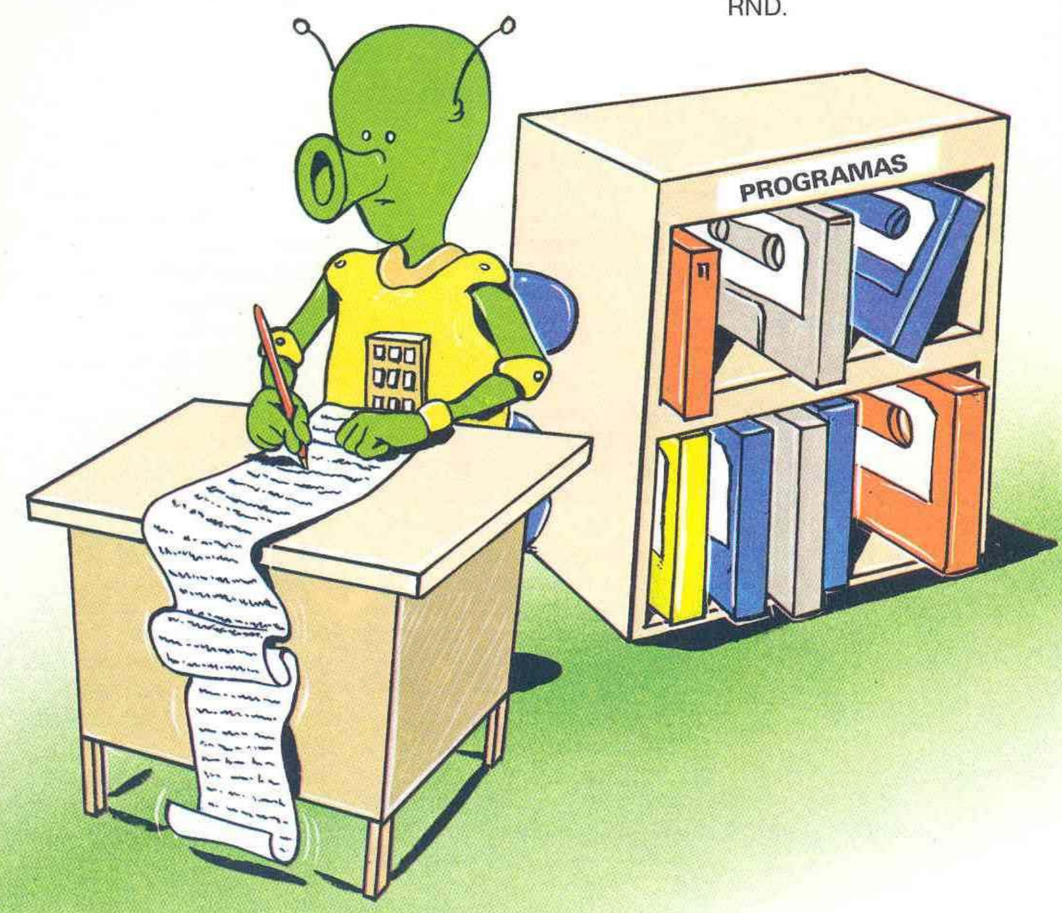
Guarda en la cinta un bloque de datos en código máquina denominada «música», a partir de la localización 29.000 y de 1.000 bytes de longitud.

```
VERIFY "música" CODE
```

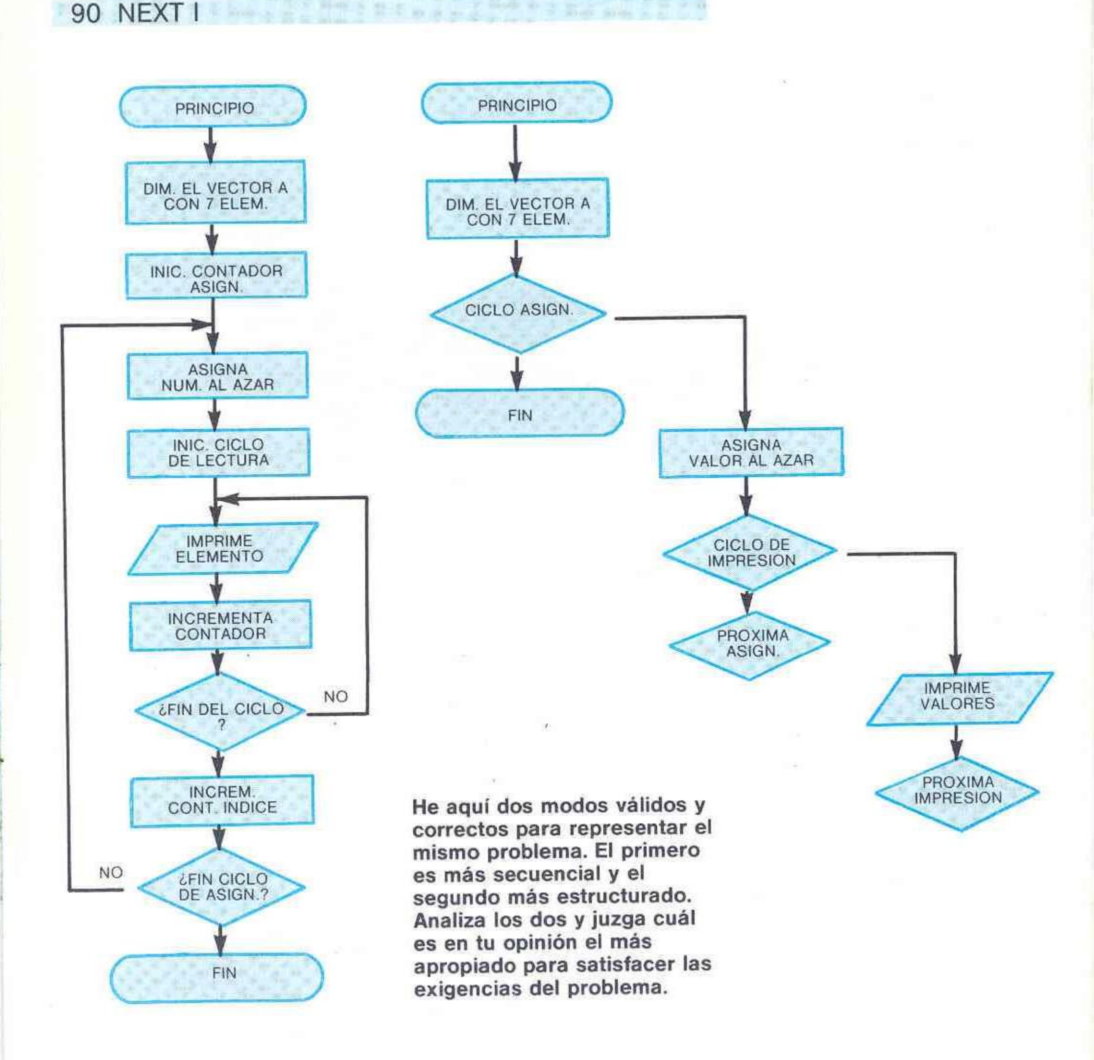
Verifica el bloque de datos anteriormente guardado.

Uso de vectores y matrices

El primer programa que pasamos a estudiar ahora es un ejemplo de introducción al uso de los vectores.
Cuando se emplean vectores es indispensable tener muy claros los conceptos de elementos, valor e índice; por lo tanto, este programa debería de ayudarte a resolver tus posibles dudas e incertidumbres.
Lo que nos proponemos hacer es bien sencillo: definiremos un vector de 7 elementos y memorizaremos en cada uno de ellos un número al azar, entre 0 y 100, generado mediante la función RND.

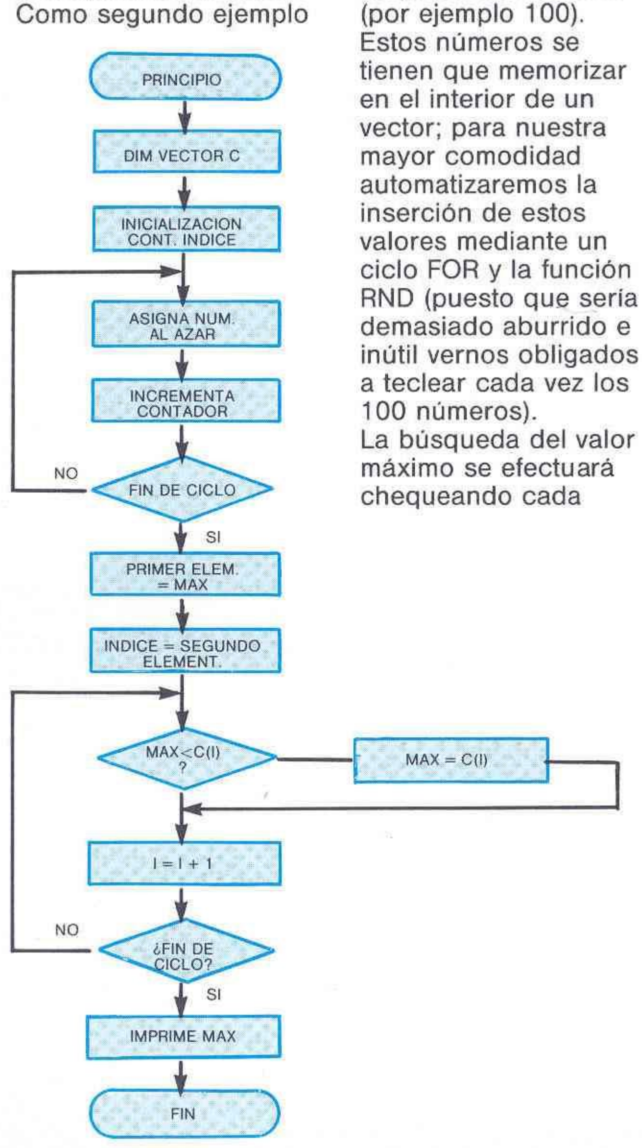


```
10 DIM A (7)
20 FOR I = 1 TO 7
30 LET A (1) = INT (RND * 100)
40 REM VISUALIZA LOS VALORES ASIGNADOS
50 FOR J = 1 TO 7: IF A (J) < 10 THEN PRINT " ";
60 PRINT A (J); "_";
70 NEXT J
80 PRINT
90 NEXT I
```



He aquí dos modos válidos y correctos para representar el mismo problema. El primero es más secuencial y el segundo más estructurado. Analiza los dos y juzga cuál es en tu opinión el más apropiado para satisfacer las exigencias del problema.

La ejecución de este programa tiene interés por la impresión repetida en pantalla de los valores contenidos por el vector.
En el interior del ciclo FOR principal (aquél que, para entendernos, empieza en la línea 20) se ha insertado un segundo ciclo (líneas 50-70) que visualiza en pantalla los valores existentes en el vector en ese momento.
A medida que la variable del índice sea incrementada, verás como los ceros —contenidos inicialmente en el vector debido a la instrucción DIM— van siendo sustituidos por números aleatorios. Por lo tanto, tendrás en tu pantalla una copia fiel de lo que vaya sucediendo en el interior de la memoria de tu ordenador durante el programa.
Como puedes observar es perfectamente ortodoxo emplear dos índices diferentes, en nuestro caso I y J, para referirse a un mismo elemento del vector: lo importante es que se respetan los límites de la definición del vector.
Como segundo ejemplo veamos ahora un programa que detecta el número máximo de un determinado conjunto de números (por ejemplo 100). Estos números se tienen que memorizar en el interior de un vector; para nuestra mayor comodidad automatizaremos la inserción de estos valores mediante un ciclo FOR y la función RND (puesto que sería demasiado aburrido e inútil vernos obligados a teclear cada vez los 100 números).
La búsqueda del valor máximo se efectuará chequeando cada



elemento del vector (asignándole a la variable MAX el valor máximo encontrado hasta ese momento) hasta que se inspeccione la totalidad del vector.

```
10 LET J = 100
20 DIM C (J)
30 REM J ES LA DIMENSION DEL VECTOR
40 REM CARGA DEL VECTOR
50 FOR I = 1 TO J
60 LET C (1) = INT (RND * 100)
70 NEXT I
80 REM BUSQUEDA DEL NUMERO
90 LET MAX = C (1): REM AL PRINCIPIO COMO MAXIMO SE
   SITUA EL PRIMER ELEMENTO
100 FOR I = 2 TO J
110 IF MAX < C (I) THEN LET MAX = C (I)
120 NEXT I
130 PRINT "EL VALOR MAXIMO ES"; MAX
140 STOP
```

Intenta modificar el programa para encontrar, además del valor máximo, el mínimo. También puede resultar interesante comprobar el drástico aumento del tiempo de ejecución debido a un incremento (por ejemplo de 100 a 500) de la dimensión del vector. Puedes experimentar todo lo que quieras.
El tercer y último ejemplo de nuestra lección es la solución al siguiente problema: dada una lista de números enteros,

encontrar cuantos números impares contiene y determina las posiciones que ocupan. Igual que antes, la inserción de los números la dejamos a RND; y también esta vez será conveniente que intentes aportar modificaciones y mejoras; por ejemplo, podrías calcular la suma de los elementos en los dos vectores, o bien, buscar en qué posiciones se encuentran los números dentro de un determinado intervalo, o, además, (aunque esto ya lo veremos en una de las lecciones siguientes) ordenar en

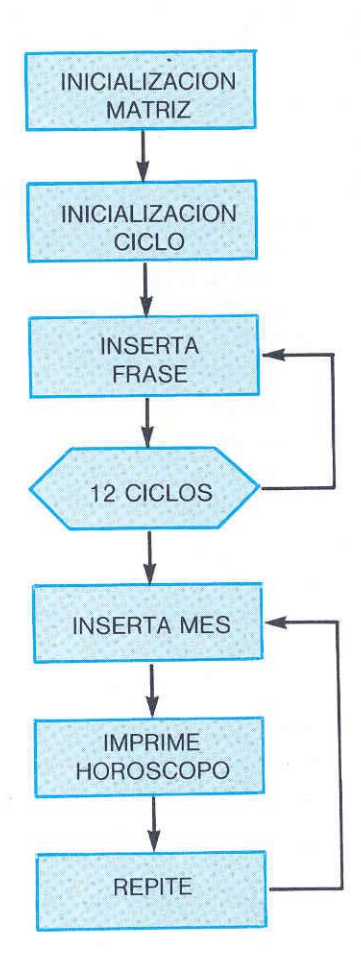
```
 10 REM EL VECTOR A CONTIENE LOS NUMEROS, P LAS
    POSICIONES DE LOS NUMEROS IMPARES
 20 INPUT "¿CUANTOS NUMEROS SERAN?": NUM
 30 DIM A (NUM): DIM P (NUM)
 40 FOR I = 1 TO NUM
 50 LET A(I) = INT (RND * 1000): REM LOS NUMEROS QUEDABAN
    COMPRENDIDOS ENTRE 0 Y 9999
 60 NEXT I
 70 REM INICILIZA EL CONTADOR DE NUMEROS IMPARES
 80 LET K = 0
 90 REM BUSCA LOS NUMEROS IMPARES
100 FOR I = 1 TO NUM
110 REM DIVIDE POR 2 EL NUMERO Y SI QUEDA UN RESTO
    SIGNIFICA QUE ES IMPAR
120 LET B = A (I)/2
130 REM TOMA LA PARTE ENTERA DE LA DIVISION
140 LET C = INT (B)
150 REM SI LA PARTE ENTERA DEL RESULTADO DE LA DIVISION ES
    IGUAL AL RESULTADO MISMO, SIGNIFICA QUE NO EXISTE PARTE
    DECIMAL Y QUE POR LO TANTO EL NUMERO ES PAR
180 IF B = C THEN GOTO 240
190 REM EL NUMERO ES IMPAR, POR LO TANTO SITUA LA POSICION
    DEL NUMERO EN EL VECTOR DE LAS POSICIONES
210 LET K = K + 1
220 LET P (K) = 1
230 REM INCREMENTA EL CONTADOR DE LOS NUMEROS IMPARES
240 NEXT I
250 REM IMPRIME LA CANTIDAD DE NUMEROS IMPARES Y SUS
    POSICIONES
270 PRINT "LOS NUMEROS IMPARES SON "; K
280 PRINT "LOS NUMEROS IMPARES OCUPAN LAS POSICIONES:"
290 FOR I = 1 TO K
300 PRINT P (I); "_";
310 NEXT I
320 STOP
```

orden creciente o decreciente los elementos, en principio colocados al azar.
Todo lo que hagas servirá para incrementar tu confianza con los vectores y las matrices.

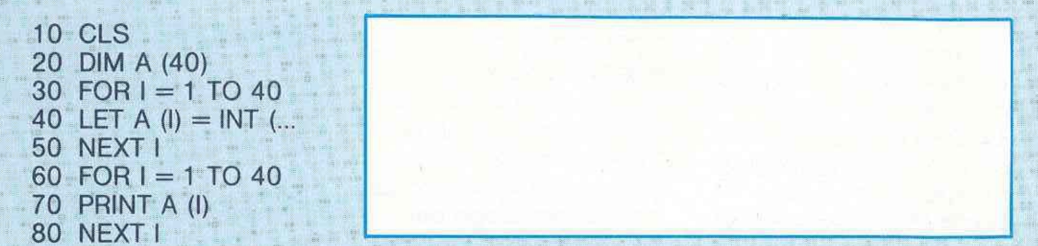
Horóscopo electrónico

El programa siguiente no es un auténtico horóscopo, más bien se trata de un juego de sociedad para aprovechar una reunión de un grupo de amigos. De cualquier manera, su efecto será proporcional a la fantasía de quien escriba las frases que aparecerán en pantalla.
Su mecanismo es el siguiente: los invitados, por turno, teclearán el número correspondiente a su mes de nacimiento, y verán aparecer su correspondiente mensaje chistoso.
Es indispensable el uso de la instrucción DIM para crear la matriz que contiene los doce textos, cada uno de los cuales tendrá un máximo de 30 caracteres.
El índice de la variable se empleará para buscar la frase.
Si lo deseas, y siempre que la capacidad de memoria lo permita, puedes modificar los valores indicados para poder introducir mensajes más largos.

```
10 DIM F$ (12,30)
20 FOR M = 1 TO 12
30 INPUT (M); F$ (M)
40 NEXT M
50 INPUT "MES    ";M
60 PRINT M'F$ (M)
70 GO TO 50
```



EJERCICIOS



```
10 CLS
20 DIM A (40)
30 FOR I = 1 TO 40
40 LET A (I) = INT (...
50 NEXT I
60 FOR I = 1 TO 40
70 PRINT A (I)
80 NEXT I
```

– Completa la línea 40 para asignar a cada elemento del vector un número entero aleatorio entre 0 y 499.

– Modifica el bucle de impresión de la línea 60, para que imprima el vector en sentido contrario.

La caza del error

```
10 CLS
20 DIM (12,10)
30 FOR I = 1 TO 12
40 INPUT D (I)
50 OR J = 1 TO 10
60 INPUT D (J)
70 NEXT I
80 NEXT J
90 REM ¡HORROR!
```

Concéntrate sobre los bucles de INPUT y piensa en los anidamientos de los ciclos FOR.
Además, ¡no es posible asignar una variable de dos dimensiones en dos tiempos!

Matriz de elementos desconocidos

```
10 INPUT "PRIMERA DIMENSION ="; P
20 INPUT "SEGUNDA DIMENSION ="; S
30 DIM C (P, S)
40 FOR I = 1 TO P
50 FOR J = 1 TO S
60 INPUT C (P, S)
70 NEXT J
80 NEXT I
90 PRINT "MATRIZ COMPLETA"
```

No exageres con las dimensiones puesto que el número de elementos a introducir lo define el producto de los índices. Añade las líneas necesarias para que impriman los valores introducidos.

SEIKOSHA SP-800

El fruto de la Investigación



La nueva impresora de **SEIKOSHA SP-800**, con un ordenador personal puede escribir ***96 combinaciones de letra diferentes***, desde 96 caracteres por segundo a 20 con muy alta calidad de letra, además es gráfica en alta densidad.

Su precio es de 69.900 Pts con ***introductor automático hoja a hoja***.

Con un pequeño ordenador personal, un procesador de textos puede costar alrededor de cien mil pesetas.

Infórmese y comprenderá por qué las **máquinas** de **escribir** tienen demasiados años.

Nuestra calidad es "SEIKO";
nuestros precios, únicos
Si desea más información,
consulte con nuestro distribuidor
más cercano, llame o escriba a:

DiRAC

DIRECCION COMERCIAL:
Av. Blasco Ibañez, 114-116
46022 VALENCIA
Tel (96) 372 88 89
Télex 62220
DIRECCION COMERCIAL EN CATALUÑA:
C/Muntaner, 60-2-4Pta
08011 BARCELONA
Tel (93) 323 32 19

ESTOS SON NUESTROS MODELOS:

MODELO	VELOCIDAD	COLUMNAS	TIPOS DE LETRA	P.V.P.R. INTERFACE PARALELO
GP-50S LA DEL SPECTRUM	40 cps	32	-	19.900
GP-50 LA PEQUEÑA	40 cps	46	2	25.900
GP-500 LA ECONOMICA	50 cps	80	2	47.900
GP-700 LA DE COLOR	50 cps	80-106	3	69.900
SP-800 LA PERFECCION	96 cps	80-137	20	69.900
BP-5200 LA DE OFICINA	200 cps	136-272	10	199.900
BP-5420 LA MAS RAPIDA	420 cps	136-272	10	299.900

* Los precios indicados son los recomendados para conexion tipo paralelo Centronics, para otro tipo de conexion,sufren un ligero incremento.

Este pie de página ha sido realizado integramente con la nueva impresora:

SEIKOSHA SP-800